

# Full Stack Development with MERN

## 1. Introduction

**Project Title:** BookEase - The Ultimate Online Bookstore

**Team Members:**

- **Team Leader :** Meka Mounika
  - **Team member:** AKKALA NAGA VEERA BHADRA KUMAR ( Full-Stack Developer)
  - **Team member :** M Naga Malleswari
  - **Team member :** Nagamani Duggempudi
- 

## 2. Project Overview

**Purpose:**

To develop a full-featured online book store that supports multi-role access (Customer, Seller, Admin), allowing customers to browse and purchase books, sellers to manage inventory, and admins to oversee platform activity.

**Features:**

- Customer registration, login, wishlist, cart, orders
  - Seller book management, analytics, and order handling
  - Admin dashboard for user, book, order, and review management
  - Review system and ratings
  - Secure role-based authentication
  - Responsive UI with Material UI
- 

## 3. Architecture

**Frontend:**

- Developed using React + Vite
- Routes managed with React Router
- Type Script

**Backend:**

- Built with Node.js and Express.js
- Middleware for authentication, authorization, error handling

# Full Stack Development with MERN

## Database:

- MongoDB (local instance or MongoDB Atlas)
  - Collections: Users, Books, Orders, Reviews, Wishlist, Cart
- 

## 4. Setup Instructions

### Prerequisites:

- Node.js ( $\geq 16.x$ )
- npm
- MongoDB Community Server (local)

### Installation:

# Clone the repo

git clone <https://github.com/bhadra0401/BookEase-WebApp-MERN.git>

cd BookEase-WebApp-MERN

# Install dependencies for client and server

cd src && npm install

cd ../server && npm install

# Setup environment

# Create `.env` in /server and add:

MONGO\_URI=mongodb://localhost:27017/Book-Ease

JWT\_SECRET=your\_jwt\_secret

---

## 5. Folder Structure

### Client (React/Vite)

src/

└─ components/

└─ pages/

# Full Stack Development with MERN

└─ contexts/

└─ stores/

└─ services/

└─ App.tsx

## Server (Node.js/Express)

server/

└─ routes/

└─ controllers/

└─ models/

└─ middleware/

└─ server.js

---

## 6. Running the Application

### Frontend:

npm run dev

### Backend:

cd server

nodemon server.js

---

## 7. API Documentation

### Auth

- POST /api/auth/register
- POST /api/auth/login

### Books

- GET /api/books/
- POST /api/books/ (seller/admin only)
- PUT /api/books/:id

# Full Stack Development with MERN

- DELETE /api/books/:id

## Orders

- GET /api/orders/my-orders
- POST /api/orders/

## Reviews

- POST /api/reviews/
  - GET /api/reviews/book/:bookId
- 

## 8. Authentication

- JWT-based auth
  - Tokens stored in localStorage
  - Middleware to protect routes and verify roles (customer, seller, admin)
- 

## 9. User Interface

- Clean, responsive layout using Material UI
  - Dark and light theme compatibility
  - Dashboards for all roles with intuitive navigation
- 

## 10. Testing

- Manual UI testing in development mode
  - Postman used for API testing
  - JWT auth flows tested across roles
- 

## 11. Screenshots or Demo

- Home Page
- Book Detail View
- Seller Dashboard (Inventory Management)

# Full Stack Development with MERN

- Admin Dashboard (User & Review Management)

GitHub Repo: <https://github.com/bhadra0401/BookEase-WebApp-MERN.git>

---

## 12. Known Issues

- No pagination on book listings
  - Admin cannot create other admins from UI
  - Review approval needs UI notifications
- 

## 13. Future Enhancements

- Add pagination and search filters
- OTP verification on user registration
- Enable real-time chat between seller and buyer
- Recommendation engine for users
- Admin review notification panel